**Amendments to the Specification:**

**The paragraph appearing on page 9, lines 14-20 is amended as follows:**

A binder is an OLE DB object that binds resources named in a URL (universal resource locator) to an OLE DB object. Binding involves associating of a column of a rowset or a parameter of a command with a variable; a binding denotes the association itself, or a structure that describes the association. Patent application Ser. No. ——————— (attorney docket 777.351US1) 09/717,533 (Attorney Docket No. MSFT-0563) also describes binders. Root binder 201 is an object that oversees the direct binding process. It maps bind requests to particular data providers such as 210. Provider binder 211 is an object that performs direct binding operations on the URL namespace for which it is registered. It creates particular objects based upon the URL specified in the bind request.

**The paragraph appearing on page 16, line 23 through page 17, line 3 is amended as follows:**

The architecture of merge subsystem 312600 312 supports the notion of a collection of merge blobs. Documents used with the invention can be explicitly typed, as mentioned earlier. A single document might carry complex information, having multiple types – for example, both content 611 and properties 612. Changes to a file during a merge could produce conflicts both in the content of the merge result and in its properties. Thus a merge collection can contain both a content merge blob 651 and a property merge blob 652. Complex types can be returned in the result, as well as a composite of other merge data. A calling program can request reports as to differences, recommendations for merging, or automatic merging. The versioning architecture allows for the promotion of version deltas, that is, moving individual changes to a file between different versions of the file in the store. Because documents from different namespaces can participate in merge operations, the architecture also manages namespace merging.

**The paragraph appearing on page 12, lines 15-26 is amended as follows:**

Client 310 interacts via versioning interface (VAPI) 310 with multiple individual providers 330 of versioning services. Interface 320 is a collection of individual API methods largely taken from the publicly available Microsoft OLE DB interface, plus methods added specifically for versioning functions. This interface forms the entire set of interactions between client ~~320~~ 310 and all of the versioning service providers 330. Alternatively, interface 320 could be fashioned from some other existing collection of API methods, could be constructed exclusively for this purpose, or could be implemented in a form other than an API. The important point is that the contract provided by the interface can be understood and followed by the client and by all the service providers, at least to the extent necessary to communicate commands and data to other modules in the same system. That is, the interface can in some cases be subsetted and still remain compatible.